

Ответы на вопросы

Вопрос 1

Что такое обучение с подкреплением?

Обучение с подкреплением - способ машинного обучения, при котором система обучается, взаимодействуя с некоторой средой. В обучении с подкреплением существует агент, который воздействует со окружающей средой, предпринимая действия. Окружающая среда дает награду агенту за его действия, а агент продолжает их предпринимать. Такой способ обучения позволяет найти компромисс между исследованием неизученных областей и применением имеющихся знаний. Среда обычно формулируется как марковский процесс принятия решений (МППР) с конечным множеством состояний, и в этом смысле алгоритмы обучения с подкреплением тесно связаны с динамическим программированием. Формально простейшая модель обучения с подкреплением состоит из:

- множества состояний окружения

$$S$$

- множества действий

$$A$$

- множества вещественных скалярных значений выигрышей

$$r$$

В произвольный момент времени

$$t$$

агент характеризуется состоянием

$$s_t \in S$$

и множеством возможных действий

$$A(s_t)$$

. Выбирая действие

$$a \in A(s_t)$$

, он переходит в состояние

$$s_{t+1}$$

и получает выигрыш

$$r_t$$

. Агент должен выработать стратегию

$$\pi : S \rightarrow A$$

, которая максимизирует величину

$$R = \sum_{i=0}^n r_i$$

в случае МППР, имеющего терминальное состояние, или величину

$$R = \sum_t \gamma^t r_t$$

для МППР без терминальных состояний (где

$$0 \leq \gamma \leq 1$$

- дисконтирующий множитель для "предстоящего выигрыша").

Вопрос 2

Как решается проблема пространственной инвариантности объекта на изображении?

Задача: выделить объекты, принадлежащие одному классу, из множества объектов, относящихся к разным классам.

Любой физический объект обладает набором некоторых свойств, которые, собственно, и позволяют отличать один объект от другого. Совокупность свойств, описывающих конкретный объект, называется образом данного объекта. Под классом объектов понимается некоторая совокупность образов, называемых элементами класса, обладающая рядом близких свойств. Измеряемые или вычисляемые свойства объектов, позволяющие отличить классы друг от друга, называются признаками. Соответственно, для решения поставленной задачи необходимо выделить эти признаки, и решить, к какому или каким классам относится выделенный объект.

При работе с изображением мы имеем дело с двумя проблемами: пространственные искажения и помехи. Чтобы решить задачу можно по возможности максимально избавиться от помех, после чего создать множество

$$M$$

состоящее из всех комбинаций выделенных преобразований каждого эталонного изображения и сравнить данное изображение со всеми изображениями множества

$$M$$

.

Такой подход имеет множество минусов: во первых, при больших размерах изображений объектов, большом числе объектов и большой мощности множества M требуется значительный объем памяти для хранения изображений, а процесс сравнения может занять очень много времени. Во вторых, при формировании множества M необходимо произвести преобразование каждого эталонного изображения при всех возможных параметрах группы. Учитывая дискретность изображения и ограниченность его размеров, число преобразований будет конечным, но для полной проективной группы может быть очень значительным. В связи с этим подобный подход реально применяется только для изображений малого размера, при наличии значительных ограничений на параметры

возможного преобразования и малом числе эталонных изображений. В остальных случаях для решения задачи распознавания объектов по их изображениям при наличии пространственных искажений возможны три основных подхода.

1. Попытаться найти признаки, инвариантные к имеющимся пространственным искажениям изображения, и по этим признакам производить сравнение образа распознаваемого объекта с эталонными образами.
2. Попытаться нормализовать изображение, устранив все имеющиеся пространственные искажения, и свести задачу распознавания объекта к последовательному сравнению изображения распознаваемого объекта в заданном положении с изображениями эталонов.
3. Комбинация 1 го и 2 го подходов, при которой устраняется часть пространственных искажений, а затем находятся признаки, инвариантные к оставшимся пространственным искажениям.

Собственно говоря, этими подходами и пользуется сверточная нейронная сеть: выделение основных признаков, инвариантных к имеющимся пространственным искажениям изображения, устранение помех, и определение класса объекта.

Вопрос 3

Что такое полностью сверточная сеть?

Устройство СНС заключается в чередовании сверточных слоев (convolutional, C-слои), реализующих концепцию простых клеток зрительной коры с гибкими (обучаемыми) весами, и субдискретизирующих слоев (sub-sampling, S-слои), построенных по примеру сложных клеток кортекса с жесткими весами. Завершается подобная каскадная модель, как правило, полносвязными нейронными слоями (fully-connected, F-слои).

В полностью сверточной сети полносвязный слой преобразуется в свертку. Для этого надо пересчитать веса полносвязного слоя в ядро свертки. При этом для эквивалентного сверточного слоя необходимо установить размерность фильтра равной размерности карты признаков, а количество фильтров должно равняться количеству нейронов данного слоя.

Пусть СНС научили определять 1000 классов. Потом полносвязный слой трансформировали в сверточный слой. Тогда эта нейросеть как бы работает тем же самым окном, на которое она была обучена (например 100x100), но теперь она может пробежаться этим окном по всему изображению и построить как бы тепловую карту на выходе — где в этом конкретном изображении находится конкретный класс.

Вопрос 4

В чем разница между методами Adam и NAdam?

По сути, Adam - это RMSProp с моментом, а Nadam - это RMSProp с моментом Нестерова. Момент Нестерова это простое изменение момента. Градиент в данном случае не вычисляется от текущей позиции

$$\theta_t$$

области параметра, а от позиции

$$\theta_{intermediate} = \theta_t + \mu v_t$$

. Это помогает, поскольку, не смотря на то, что градиент всегда направлен в верном направлении, момент может быть направлен неверно. Если момент направлен в неверном направлении, или проскакивает, градиент все равно может "вернуться" и исправить это в том же шаге обновления.

Вопрос 5

Почему используется кросс-энтропия?

Кросс-энтропия используется для определения функции потерь в задаче определения классов. На выходе модели у нас имеется набор предсказанных вероятностей. Заметим, что выходной нейрон в модели отвечает только за один класс. Если на входе модели объект того класса, за который отвечает нейрон, мы желаем видеть единицу на его выходе, в противном случае — ноль. В реальном же предсказании класса, искусственный нейрон активируется в открытом диапазоне между нулем и единицей, при этом значение может быть сколь угодно близким к этим двум асимптотам. Значит, чем точнее мы угадываем класс, тем меньше абсолютная разница между реальным классом и активацией нейрона, отвечающего за этот класс.

Пусть

$$p$$

— это правильные метки (ожидаемый результат), и

$$q$$

— это предсказанные моделью метки. Определив нотацию

$$p \in \{y, 1 - y\}$$

и

$$q \in \{\hat{y}, 1 - \hat{y}\}$$

мы можем использовать кросс-энтропию, чтобы измерить разницу между

$$p$$

и

$$q$$

:

$$H(p, q) = - \sum_i p_i \log(q_i) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$$

Логистическая регрессия обычно оптимизирует логарифмическую потерю для всех наблюдений, на которых она тренировалась, что по сути то же самое, что оптимизация среднего кросс-энтропии в пробе. Например, предположим, что у нас имеется

$$N$$

проб, каждая проба проиндексирована как

$$n = 1, \dots, N$$

. Среднее от функции потерь определяется как:

$$J(w) = \frac{1}{N} \sum_{n=1}^N H(p_n, q_n) = -\frac{1}{N} \sum_{n=1}^N [y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n)]$$

С другой стороны на это можно смотреть так: попробуем создать функцию потерь, которая бы возвращала числовое значение штрафа, такое, чтобы оно было маленьким в том случае, когда нейросеть выдает значения близкие к значению класса, и очень большим в том случае, в котором нейросеть выдает значения, приводящие к неправильному определению класса.

С этим прекрасно могут разобраться функции:

$$-\log h$$

и

$$-\log(1 - h)$$

.

Теперь осталось записать функцию потерь в виде выражения.

$$J(w) = -\frac{1}{m} \sum_{n=1}^m (y^{(i)} \log(h_w(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_w(x^{(i)})))$$

Как можно увидеть, эта формула совпадает с логарифмической (кросс-энтропийной).